第3章

タッチ・パネル付き TFTカラー液晶表示回路を作ろう

位置情報の取得方法とバックライト用LED駆動回路の設計

江崎雅康/森川聡久

組み込み機器の開発において，LEDや7セグメント表示器以外に，文字情報やグラフィック情報を表示したいときがある．表示のためにパソコンを用意するのではなく，カラー液晶表示モジュールに直接表示させようというのが今回のもくろみである．（編集部）

カラー液晶表示モジュールには，大別してSTN方式とTFT方式があります．TFT方式は表示ピクセルごとに駆動トランジスタを配置した構造になっています．コスト高ですがきれいなカラー画像を表示できます．

バックライト光源には冷陰極管（CFL：cold-cathode fluorescent lamp）もしくはLEDが用いられています．携帯電話など超小型機器用の液晶にはLEDを利用した品種が多いようです．

写真1　タッチ・パネル付きTFTカラー液晶表示モジュールTCG057VGLAD-G00（京セラ）
公称5.7インチVGA（640×480），262,144色カラー表示可能．左側に伸びている4芯FPC（flexible printed circuits）はタッチ・パネル接続用ケーブル．背面に回りこんでいるFPC（33芯）は信号インターフェース用ケーブル．下部に見えるツイスト線はバック・ライト駆動用．

1. TFTカラー液晶モジュールTCG057QVGLAD-G00の仕様

…VGA（640×480），262,144色のカラー表示が可能

写真1は画像ベースボード用に用意したTFTカラー液晶表示モジュール「TCG057VGLAD-G00」（京セラ）です．

表1は製品仕様です．公称5.7インチVGA（640×480）サイズの画面に262,144色のカラー表示ができます．

表2はインターフェース信号配列です．同サイズQVGA（320×240）モデル「TCG057QVGLAD-G00」（京セラ）と同じ信号配列です．

バックライトは冷陰極管モデルとLEDモデルがあります．製品としては冷陰極管モデルの方が多く製造されているようですが，今回はLEDモデルを採用しました．

表3はバックライトLED接続用コネクタのピン配列で

表1[1]　TFTカラー液晶モジュールTCG057VGLAD-G00の仕様

項　目	仕　様	単　位
外形寸法	144.0（*W*）×104.8（*H*）×14.8（*D*）	mm
有効表示領域	117.2（*W*）×88.4（*H*）	mm
ドット構成	（640×R.G.B）（*W*）×480（*H*）	ドット
ドット・ピッチ	0.06（*W*）×0.18（*H*）	mm
表示モード注1	ノーマリ・ホワイト	―
質量	265	g

注1：LCDパネルの色調は，LCDパネルの特性として環境温度により変化する．

KeyWord　タッチ・パネル付きTFTカラー液晶表示モジュール，TCG057QVGLAD-G00，昇圧レギュレータ，LM2733，マイクロコントローラ，ADuC7026

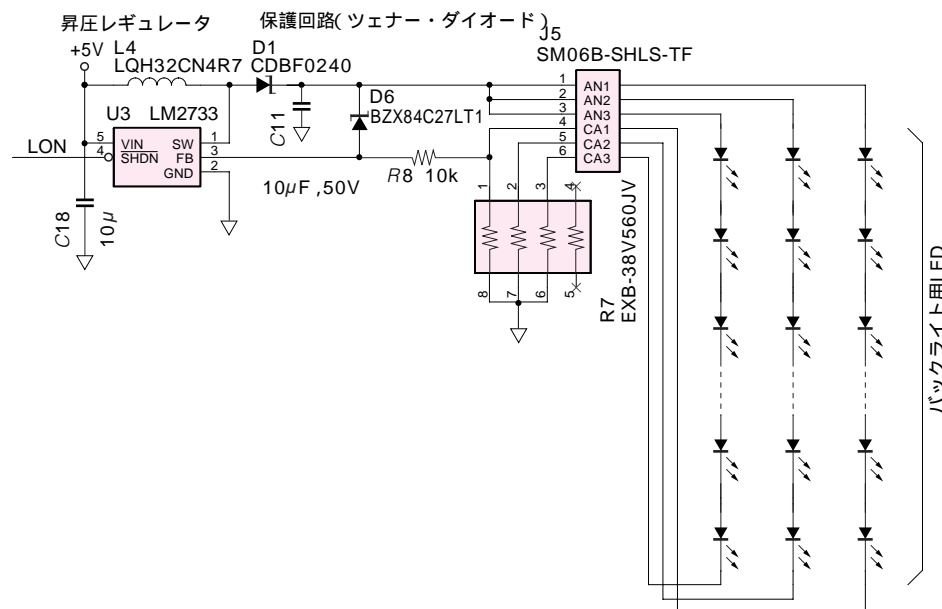



図1
液晶のバックライト用LED駆動回路
順方向電圧22.1～24.2V，電流25mAのバックライト用LEDモジュールを3組まとめて駆動する．

表2[1]
TFTカラー液晶表示モジュールTCG057VGLAD-G00のインターフェース信号配列
TFT液晶表示モジュールのインターフェース信号は水平/垂直同期信号とピクセル・クロック（PCLK），カラー画像データ・バス（RGB3色×6ビット）が標準になっている．信号配列やタイミング仕様はメーカによって異なる．京セラのこの製品は3.3V単電源で使いやすい．

使用コネクタ：6210-033（京セラエルコ）
適用FFC：FFC またはFPC（0.5mmピッチ）

ピン番号	記号	内容	I/O	備考
1	GND	グラウンド	—	
2	CK	データ・サンプリング・クロック信号	I	
3	Hsync	水平同期信号（負極性）	I	
4	Vsyne	垂直同期信号（負極性）	I	
5	GND	グラウンド	—	
6	R0	赤データ信号（LSB）		
7	R1	赤データ信号	I	
8	R2	赤データ信号	I	
9	R3	赤データ信号	I	
10	R4	赤データ信号	I	
11	R5	赤データ信号（MSB）	I	
12	GND	グラウンド	—	
13	G0	緑データ信号（LSB）	I	
14	G1	緑データ信号	I	
15	G2	緑データ信号	I	
16	G3	緑データ信号	I	
17	G4	緑データ信号	I	
18	G5	緑データ信号（MSB）	I	
19	GND	グラウンド	—	
20	B0	青データ信号（LSB）	I	
21	B1	青データ信号	I	
22	B2	青データ信号	I	
23	B3	青データ信号	I	
24	B4	青データ信号	I	
25	B5	青データ信号（MSB）	I	
26	GND	グラウンド	—	
27	ENAB	水平表示位置信号（正極性）	I	注1
28	V_{DD}	電源入力（+3.3V）	—	
29	V_{DD}	電源入力（+3.3V）	—	
30	R/L	左右反転信号（L：通常，H：左右反転）	I	
31	U/D	上下反転信号（H：通常，L：上下反転）	I	
32	V/Q	VGA，QVGA切り替え信号（H：通常（VGA））	I	
33	GND	グラウンド	—	

注1：水平位置はENAB信号の立ち上がりで規定される．ENABが“L”固定の場合はモジュール内で設定された表示位置で規定される．“H”固定では使用しない．

表3[1] **TFTカラー液晶モジュールTCG057VGLAD-G00のバック・ライト用LEDを接続するためのコネクタのピン配列**
バックライト用LEDは「アノード-カソード」ラインが3系統ある．3系統とも点灯しないと画面が均一な明るさにならない．

ピン番号	記号	内容
1	AN1	アノード1
2	AN2	アノード2
3	AN3	アノード3
4	CA1	カソード1
5	CA2	カソード2
6	CA3	カソード3

使用コネクタ：SHLP-06V-S-B（日本圧着端子製造）
適合コネクタ：SM06B-SHLS-TF（日本圧着端子製造）
：SM06B-SHLS-TF（LF）（SN）
…RoHS対応コネクタ
（日本圧着端子製造）

す．**図1**に示すように白色LEDを直列に接続したLED群が3列，組み込まれています．

オーダ時のオプション仕様としてタッチ・パネルの有無を指定できます．当然コストに上乗せされますが，今回の設計は「できる限り多くの要素技術を試す」ことを目標にしているので，タッチ・パネル付きモデルを使うことにしました．

表4はタッチ・パネル・コネクタの信号配列と仕様です．

2. TFTカラー液晶表示モジュールの構成

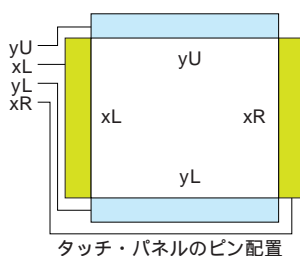
図2はTFTカラー液晶表示モジュールTCG057VGLAD-G00の構成図です．モジュールの厚さは14.8mmですが，この中に，

- 液晶パネル
- カラー・フィルタ
- 偏光板
- ドライバIC
- LEDバック・パネル

表4[1] **TFTカラー液晶モジュールTCG057VGLAD-G00のタッチ・パネル用コネクタの信号配列と仕様**

番号	記号	信　号
1	yU	y－上側端子
2	xL	x－左側端子
3	yL	y－下側端子
4	xR	x－右側端子

タッチ・パネルのピン配置

使用FFC　：1.25mmピンチ
適合コネクタ：6216（京セラエルコ）
FE，FFS
（日本圧着端子製造）
SFD（フランスFCI社）

（a）信号配列

1. 端子間抵抗	xL～xR：200Ω～1000Ω yU～yL：200Ω～1000Ω
2. 直線性	x方向：1.5%以下 y方向：1.5%以下
3. 絶縁抵抗	DC25V　100MΩ以上

（b）仕様

表5[1] **TFTカラー液晶モジュールTCG057VGLAD-G00の電気的特性**
電源系統は3.3Vで統一されている．消費電流210mA（標準）にはバック・ライト分は含まれない．

項　目	記　号	最　小	最　大	単　位
入力電源電圧	V_{DD}	0	4.0	V
入力信号電圧注1	V_{in}	－0.3	6.0	V
タッチ・パネル電源電圧	V_{tp}	0	6.0	V
接点通過電流	I_{tp}	0	0.5	mA

注1：入力信号：CK，R0～R5，G0～G5，B0～B5，Hsync，Vsync，ENAB，R/L，U/D，V/Q

（a）電気的絶対最大定格

T_a=－10～70℃

項　目		記　号	最　小	標　準	最　大	単位
電源入力電圧注2		V_{DD}	3.0	3.3	3.6	V
消費電流	V_{DD}=3.3V T_a=25℃	I_{DD}	－	210	270	mA
許容入力リプル電圧	V_{DD}=3.3V	V_{RP}	－	－	100	$mV_{p\text{-}p}$
入力電圧（Low）		V_{IL}	0	－	$0.3V_{DD}$	V
入力電圧（High）		V_{IH}	$0.7V_{DD}$	－	V_{DD}	V

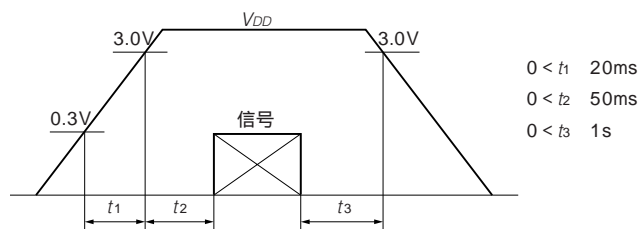
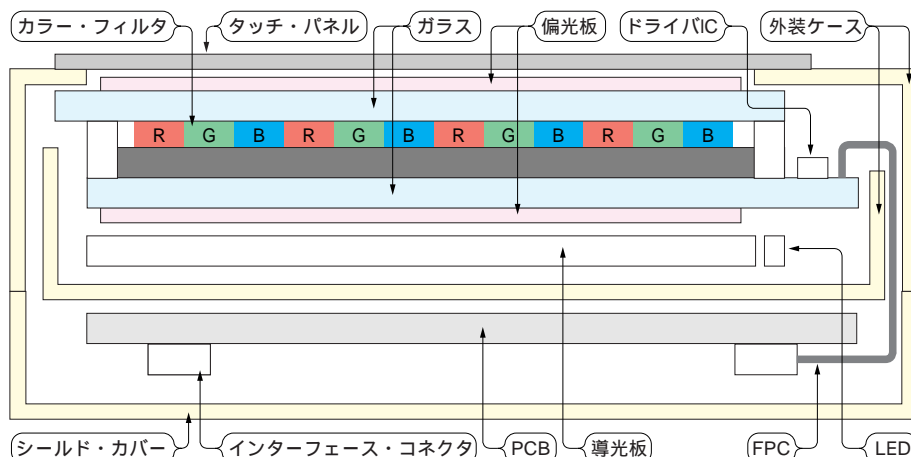
注2：入力電源シーケンス

（b）電気的特性

図2[1]
TFTカラー液晶モジュールTCG057VGLAD-G00の構成図
液晶パネル，カラー・フィルタ，偏光板，ドライバIC，LEDバック・パネル，タッチ・パネルがぎっしりと実装されている．厚さ14.8mm．

● タッチ・パネル

が実装されています．**表2**のインターフェース信号のうち

● 左右反転信号　　R/L
● 上下反転信号　　U/D

は**図3**に示すように画面表示をコントロールします．

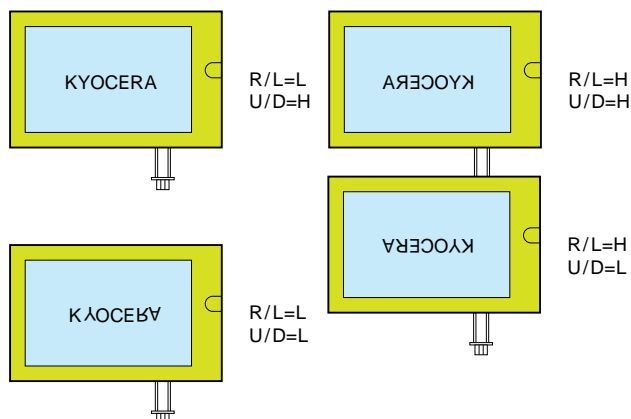



図3[1]　**TFT カラー液晶モジュールTCG057VGLAD-G00 の画面制御信号 R/L（左右反転信号），U/D（上下反転信号）の仕様**

左右反転信号，上下反転信号により液晶モジュールの画面表示方向を切り替えることができる．

3. TFT カラー液晶表示モジュールのインターフェース信号

表5はTFT カラー液晶表示モジュールTCG057VGLAD-G00 の電気的特性です．バックライト光源を除けば3.3V 単電源で駆動できるので回路設計が容易です．

図4はTCG057VGLAD-G00 の入力タイミング図，**表6**は各タイミング・パラメータです．このタイミングはCRT 表示装置を踏襲しているので，

● 水平同期信号　　Hsync
● 垂直同期信号　　Vsync

などの同期信号は第2章（pp.29-38）で紹介したアナログRGB 表示装置とまったく同じです．

液晶表示装置には「水平帰線期間」「垂直帰線期間」は必要ないはずですが，**図4**のタイミング図はこの帰線期間を含んでいます．

水平表示位置信号ENAB はブランキング信号nLBLANK とほぼ同じ信号です．**図4**のコメントにあるようにENAB を"L"固定にすると標準的なフロント・ポーチ期間が挿入されます．しかし的確なENAB 制御を行えば時間的にむだなフロント・ポーチを省くことができます．

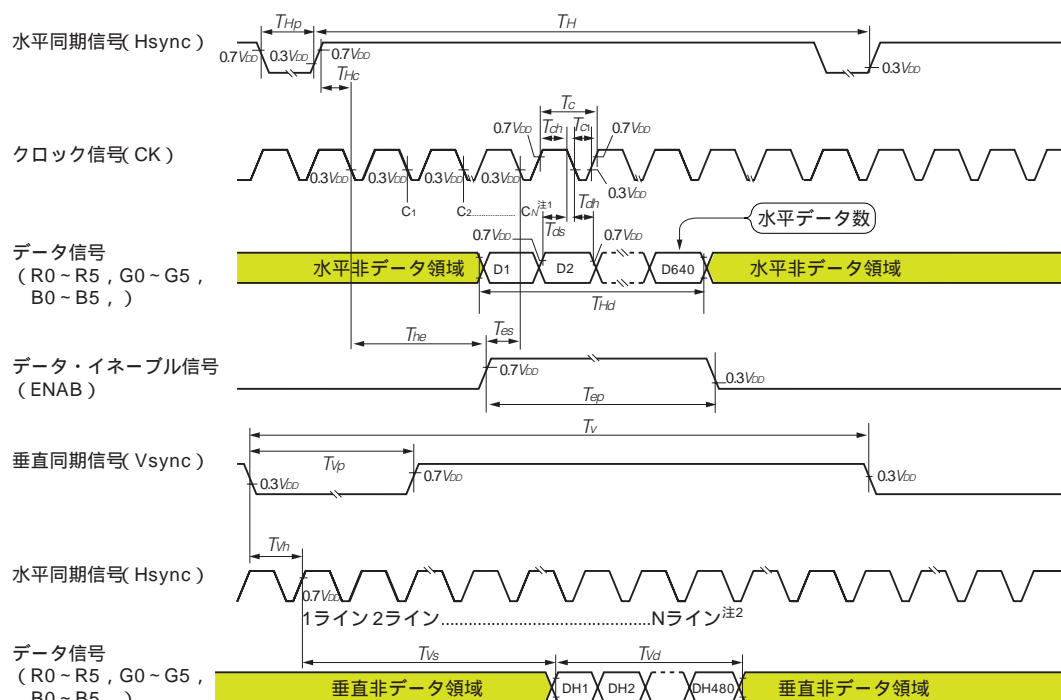



図4[1]
TFT カラー液晶モジュール TCG057VGLAD-G00 の入力タイミング図

水平/垂直同期信号，データ・イネーブル信号はアナログRGB 表示装置の信号がそのまま使える．クロック信号（CK）の立ち下がりエッジで画像データが取り込まれる．

データ・サンプリング・クロック(CK)は，ピクセル・クロックに相当する信号です．**図4**に示すように，データ・サンプリング・クロック(CK)に同期して，

- 赤データ信号　LR[0：5]
- 緑データ信号　LG[0：5]
- 青データ信号　LB[0：5]

を出力します．

この18ビットのピクセル・データは**図5**に示すように，CRT表示装置と同様にドット表示されます．

4．アナログRGB表示回路の信号がそのまま使える液晶表示インターフェース回路の設計

図6はTFT液晶モジュールTCG057VGLAD-G00のインターフェース回路です．なんと第2章のアナログRGB表示回路の信号がそのまま使えました．

試作段階ではピクセル・データのビット数を5ビットに絞って

- ブランキング信号(nLBLANK)と水平表示位置信号(HENB)
- ピクセル・クロックとデータ・サンプリング・クロック

に別々のピンを割り当てました．

しかしTCG057VGLAD-G00に限ればタイミング的に同じ信号でかまわない…という結論になりました．**写真2**はブロックくずしのデモ画面を液晶モジュールに表示したところです．

バックライト回路とタッチ・パネルは，液晶モジュール固有のものですから，これは独自に設計せざるを得ません．

5．バックライトLED駆動回路の設計

表7はTCG057VGLAD-G00のバックライト・システムの仕様です．**図1**に示した3組のLEDモジュールに標準25mAの電流を流す必要があります．LEDモジュールの順方向電

D1, DH1	D2, DH1	D3, DH1	・・・・・・・・・	D640, DH1
D1, DH2	D2, DH2	D3, DH2		
	・		RGB	
D1, DH480	D2, DH480	D3, DH480		

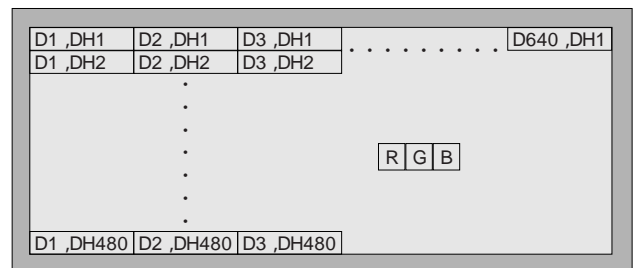



図5　TFTカラー液晶モジュールTCG057VGLAD-G00のデータの画面表示位置
アナログRGBディスプレイと同じ順序で画像ピクセルが並ぶ．

表6[1]
TFTカラー液晶モジュールTCG057VGLAD-G00の入力信号タイミング特性

項目		記号	最小	標準	最大	単位
クロック	周波数	$1/T_c$	—	25.18	28.33	MHz
	デューティ比	T_{ch}/T_c	40	50	60	%
データ	セットアップ・タイム	T_{ds}	5	—	—	ns
	ホールド・タイム	T_{dh}	10	—	—	ns
水平同期信号	周期	T_H	30.0	31.8	—	μs
			770	800	900	クロック
	パルス幅	T_{Hp}	2	96	200	クロック
垂直同期信号	周期	T_V	515	525	560	ライン
	パルス幅	T_{yp}	2	—	34	ライン
水平表示範囲		T_{Hd}		640		クロック
水平同期信号-クロック位相差		T_{Hc}	10	—	$Tc-10$	ns
水平-垂直同期信号位相差		T_{Vh}	0	—	T_H-T_{Hp}	ns
垂直データ開始位置		T_{Vs}		34		ライン
垂直表示範囲		T_{Vd}		480		ライン

注：周波数が遅くなると，フリッカなど表示品位の低下をまねく場合がある

(a) タイミング特性

項目		記号	最小	標準	最大	単位
イネーブル信号(ENAB)	セットアップ・タイム	T_{es}	5	—	T_c-10	ns
	パルス幅	T_{ep}	2	640	T_H-10	クロック
水平同期信号とイネーブル信号の位相差		T_{he}	44	—	104	クロック

(b) 水平表示位置（水平表示位置はイネーブル信号(ENAB)の立ち上がりで規定される）

写真2　VGA液晶画面にブロックくずしゲームを表示

昔なつかし「ブロック崩しゲーム」をFPGAで再現．ボールが楕円形に写っているのはボール速度が速いため？（撮影したデジカメのシャッタ速度が遅い）

図6　TFT液晶インターフェース回路

画像データおよび同期信号はアナログRGB回路の信号がそのまま使える．Spartan-3Eの出力信号を直結．液晶画面モード信号：R/L,U/D,V/QはマイクロコントローラADuC7026のGPIOにより設定．J9はタッチ・パネル接続コネクタ．

圧はLEDの個体差および温度特性により変動します．

順方向駆動電圧22.1～24.2V（標準値）は画像ベースボード上で作ります．米国National Semiconductor社の昇圧レギュレータ「LM2733」を使って＋5Vから昇圧します．

LM2733は**図7**に示すようにSOT-23パッケージに入っ

表7[(2)]　TFTカラー液晶モジュールTCG057VGLAD-G00のバックライト・システムの仕様

バックライト用LEDの定格電流は各25mA．流す電流量により画面の明るさをコントロールできる．LEDの順方向電圧は温度によって変わる・また個体差もあるので定電流駆動もしくは電流制限抵抗を直列に入れた定電圧駆動が必要．

項目	記号	標　準	最大	単位	備考
順方向電流注1	I_F	25	—	mA	T_a=－10～+70℃
順方向電圧	V_F	24.2	27.0	V	I_F=25mA 注1, T_a=−10℃
		23.1	25.9	V	I_F=25mA 注1, T_a=+25℃
		22.1	24.9	V	I_F=25mA 注1, T_a=+70℃
寿命注2	T	50,000注3	—	V	I_F=25mA 注1

T_a=25℃

注1：AN1-CA1，AAN2-CA2およびAN3-CA3間のそれぞれについて
注2：表面輝度が初期の50%に減じたとき
注3：寿命は予測値

図7[(2)]　昇圧レギュレータLM2733のピン配置とパッケージ

3mm角足らずの小さなSOT-23パッケージ．

表8[(2)]　バックライト用LEDの駆動に採用したLM2733の端子機能

端子番号	端子名	機能
1	SW	内部FETスイッチのドレイン．
2	GND	アナログ回路とパワー回路のグラウンド．
3	FB	外付け抵抗分圧回路に接続する帰還入力．
4	SHDN	シャットダウン制御入力．シャットダウン機能を使用しない場合はV_{IN}に接続する．
5	V_{IN}	アナログ回路とパワー回路の電源入力．

た小さなICです．**表8**はLM2733の端子機能です．**図8**の内部ブロックに示すように，出力段のドライバFETおよび電流検出抵抗まで入ったたいへん高機能なICです．

表9はLM2733の電気的特性です．

- スイッチング電流　最高1A
- 出力電圧　最高40V
- スイッチング周波数　300kHz～1.6MHz

と，25mAのLEDを駆動するのには少し贅沢なICです．

図1の回路はLM2733を使ったおかげでたいへんシンプルな回路ですが，ここにいたるまでには少し試行錯誤がありました．

● 3組のステップアップ回路を簡略化

最初の試作基板ではLM2733を3個使って各LEDモジュールごとに電流帰還をかけて正確に25mAが流れるような設計にしました．

ところがコストを計算してみてびっくりしました．LM2733だけでなく，

- パワー・ショットキ・ダイオード　CDBF0240
- パワー・インダクタ　LQH32CN2R7

図8[2] 昇圧レギュレータLM2733の内部ブロック図
出力段のドライバFETを内蔵し使いやすい．スイッチング電流1A（最大），出力電圧40V（最大）．

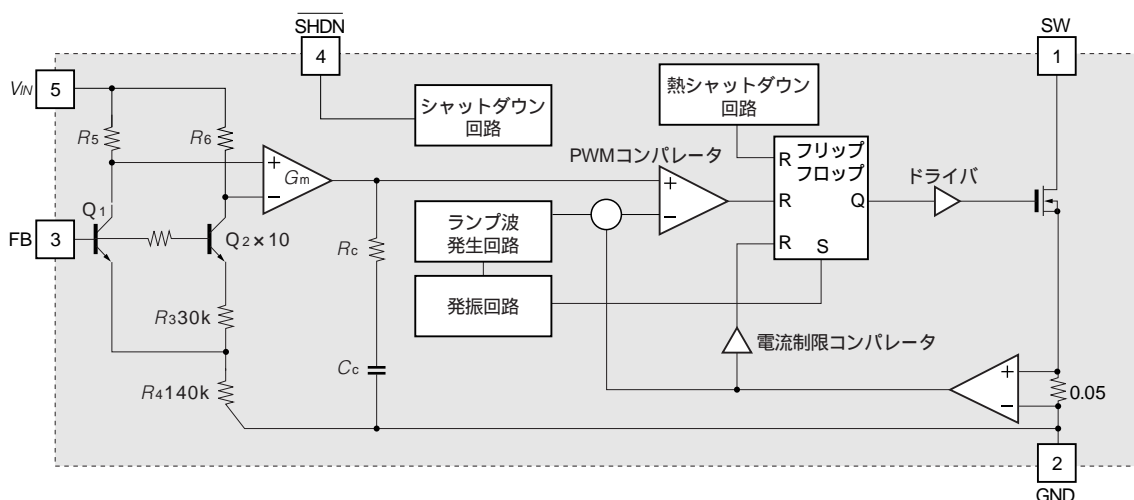



表9[2] 昇圧レギュレータLM2733の電気的特性

記　号	パラメータ	条　件	最　小	標　準	最　大	単位
V_{IN}	入力電圧		**2.7**		**14**	V
I_{SW}	スイッチング電流制限値		1.0	1.5		A
$R_{DS(ON)}$	スイッチングON抵抗	I_{SW}＝100mA		500	650	mΩ
$SHDN_{TH}$	シャットダウン電圧値	デバイスON	**1.5**			V
		デバイスOFF			**0.50**	
I_{SHDN}	シャットダウン・ピン・バイアス電流	V_{SHDN}＝0		0		μA
		V_{SHDN}＝5V		0	**2**	
V_{FB}	フィードバック・ピン基準電圧	V_{IN}＝3V	**1.205**	1.230	**1.255**	V
I_{FB}	フィードバック・ピン・バイアス電流	V_{FB}＝1.23V		60		nA
I_Q	静止電流	V_{SHDN}＝5V，スイッチング“X”		2.1	**3.0**	mA
		V_{SHDN}＝5V，スイッチング“Y”		1.1	**2**	
		V_{SHDN}＝5V，非スイッチング時		400	**500**	μA
		V_{SHDN}＝0		0.024	1	
ΔV_{FB} / ΔV_{IN}	FB電圧ライン・レギュレーション	2.7V≦V_{IN}≦14V		0.02		%V
F_{SW}	スイッチング周波数	“X”オプション	**1.15**	1.6	**1.85**	MHz
		“Y”オプション	**0.40**	0.60	**0.8**	
D_{MAX}	最大デューティ・サイクル	“X”オプション	**87**	93		%
		“Y”オプション	**93**	96		
I_L	スイッチもれ電流	非スイッチングV_{SW}＝5V			1	μA

標準書体のリミット値はT_J＝25 ℃に対して適用され，太字のリミット値は全動作温度範囲（－40 ℃≦T_J≦＋125 ℃）で適用される．特記のない限り，以下の規格は，V_{IN}＝5V，V_{SHDN}＝5V，I_L＝0mAの場合に適用される

●積層チップ・コンデンサ　　10 μF　耐圧50V
などの部品も3組必要になります．

10μFの積層チップ・コンデンサも耐圧50Vとなると結構コストアップになります．

そこでコストダウンのため，正確なレギュレータ制御による定電流回路は1組だけにしました．残りの2組のLEDモジュールは電流制限抵抗による定電圧駆動としました．

3組のLEDモジュールは同じ温度環境におかれると考えられます．**表7**の仕様のうち考慮すべきは順方向電流の個体差で，約10%程度と考えられます．

定格25mAの駆動電流が22.5mAになっても27.5mAになっても大きな支障はないと考えて図のような回路に変更しました．LM2733の駆動能力は十分あるので3組駆動しても問題ありません．

● 開放保護回路

実は最初の試作基板では，ツェナ・ダイオードD6，抵抗*R*8で構成する保護回路を備えていませんでした．

LM2733のFB（フィードバック）端子をLEDの電流検出抵抗に直結していました．

基板にバックライト用LEDのコネクタを接続しない状態で電源を入れるとLM2733はLEDモジュールに電流を流そうと出力端の電圧をどんどん上げます．

そしてついにドライバFETの絶対最大定格電圧の40Vを超えてFETを破壊してしまいます．第1次試作基板では幾度となく基板から白い煙が出ました．

図1の回路は大丈夫です．LM2733の出力端の電圧が29V前後になるとツェナ・ダイオードD6と抵抗*R*8に電流が流れFB端子の電圧が上昇して出力端の昇圧を停止させます．

図9　タッチ・パネル付き液晶表示システムの構成図
タッチ・パネルの押された*XY*座標をマイクロプロセッサADuC7026で読み取ってシリアル出力する．

6．画像ベースボードによるタッチ・パネルの座標読み取り

画像ベースボードCQ-SP3EDWには，タッチ・パネル液晶との接続端子J9が搭載されています．タッチ・パネル液晶のタッチ位置の座標を画像ベースボードに搭載されているマイクロコントローラADuC7026で読み取る方法について紹介します．

写真1で左側にまっすぐ突き出している4芯のFPCケーブルがタッチ・パネルの接続端子です．これを画像ベースボードのコネクタJ9に接続します．線の白い面が上になるように注意して接続します．

タッチ・パネルの座標読み取りはADuC7026のみで行います．7月号付属FPGA基板は直接使用していないので画像ベースボードに搭載していなくても動作します（搭載していても問題はない）．

今後，カメラとタッチ・パネル液晶との連携を行う際には搭載する必要があります．

7．ソフトウェア開発に使ったツール

今回使用したソフトウェア開発ツールは，
●統合開発環境 Keil μVision3
●フラッシュROMライタ ARMWSD
です．

これらのツールは，開発したサンプル・プログラムを動作させる際に必要になるので，あらかじめパソコンにインストールしておきます．なお，使い方に関してはここでは割愛します．必要に応じて本誌2006年3月号を参照ください．

8．タッチ・パネル付き液晶表示システムの構成

今回開発したシステムの概要を**図9**に示します．タッチ・パネル液晶表示モジュールからは，TOP，BOTTOM，LEFT，RIGHTの4本の端子が出ています．このうちTOPとLEFTの電圧を取得することで，タッチしている場所を知ることができます．電圧の値はADuC7026のA-Dコン

バータの機能を利用して，ディジタル・データに変換して取得します．

ADuC7026側では，

(1) A-Dコンバータからデータの取得
(2) 座標データに変換
(3) 座標データをシリアル出力

を定期的に行います．

9．タッチ・パネルの原理とタッチ位置の取得方法

タッチ・パネルのタッチ位置を検出する原理には，

- 抵抗膜方式
- 静電容量方式
- 超音波表面弾性波方式
- 赤外線遮光方式
- 電磁誘導方式
- 画像認識方式

など多くの方式があります．

TCG057VGLAD-G00のタッチ・パネルは抵抗膜方式です．**図10**は抵抗膜方式タッチ・パネルの動作原理です．横方向に抵抗膜R_X，縦方向にR_Yが配置されています．

抵抗膜R_XおよびR_Yから縦横に無数の導電スリット・パターンが配置された構造になっています．図のポイントAが押されると縦横のパターン間が導通し，右側に示す等価回路が生成できます．

図11はタッチした際の抵抗膜の分圧比からX座標，Y座標を算出する手順を示したものです．

まず同**図(b)**に示すように抵抗膜R_Xの両端に電圧をかけます．抵抗膜R_Yはフローティング状態にして，その電圧を測ることにより，

R_{X1}：R_{X2}

の比を求めます．この値から**図10**のX座標が求まります．座標検出の精度を高めたい時は**図11(b')**に示すように抵抗膜にかける電圧の向きを逆にして分圧比を求め平均値を算出します．

同様にして**図11(c)**に示すように抵抗膜R_Yの両端に電圧をかけてY座標を求めます．

10．画像ベースボードのタッチ・パネル・インターフェース

タッチ・パネルの座標検出には専用のICも市販されていますが，画像ベースボードではマイクロコントローラADuC7026のGPIO端子とアナログ入力端子を活用します．

まず電圧をかける方法ですが，

- ＋電圧　　GPIOを出力に設定して“H”を出力
- －電圧　　GPIOを出力に設定して“L”を出力

により行います．

この時，電圧計測側の端子は，

- 高インピーダンス　　GPIOを入力に設定

とし，アナログ入力端子の電圧をA-Dコンバータで読み取

図10　抵抗膜方式タッチ・パネルの動作原理
押された場所のXY座標が抵抗膜の分圧値として検出できる．

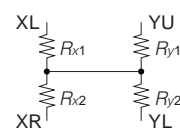
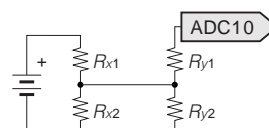
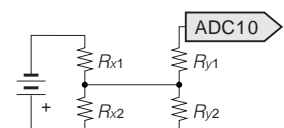
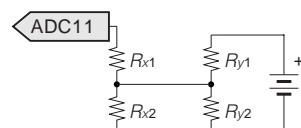
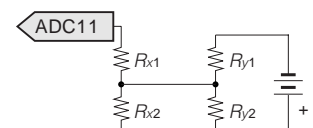



図11　四つの抵抗膜R_{X1}，R_{X2}，R_{Y1}，R_{Y2}を等価回路で表した図
XY座標の読み取りは抵抗膜に掛けた電圧の分圧値として読み取る．

表10　タッチ・パネルの端子接続と入出力設定

タッチ・パネル端子	ADuC7026端子	入出力設定	出力値
TOP	P0.3	入力	―
BOTTOM	P0.4	入力	―
LEFT	P0.5	出力	0
RIGHT	P0.6	出力	1

(a) 左右の座標を読み取る時の入出力設定

タッチ・パネル端子	ADuC7026端子	入出力設定	出力値
TOP	P0.3	出力	0
BOTTOM	P0.4	出力	1
LEFT	P0.5	入力	―
RIGHT	P0.6	入力	―

(b) 上下の座標を読み取る時の入出力設定

31 ～ 24	23 ～ 16	15 ～ 8	7 ～ 0
入出力方向 0:入力,1:出力	出力データ	リセット時に ピンの状態を反映	入力データ

※各設定とも，最上位ビットは7ピン，最下位ビットは0ピン．

図12　マイクロコントローラADuC7026のGP0DATレジスタ

表11　マイクロコントローラADuC7026のA-Dコンバータ関連のレジスタ一覧

レジスタ名	内　容
ADCCON	ADCに関する各種設定
ADCCP	正チャネルの選択
ADCCN	負チャネルの選択(シングル・エンド・モードでは使用しない)
ADCSTA	変換完了かを示す
ADCDAT	変換結果を保持
ADCRST	ディジタル・インターフェースをリセット
ADCGN	ゲイン・キャリブレーション
ADCOF	オフセット・キャリブレーション
REFCON	バンドギャップ・リファレンスの設定

リスト1　A-Dコンバータの設定

```
void AD_init(void)
{
    int i = 0;
    ADCCON = 0x0020;          // A-Dコンバータの電源ON
    while (i <= 20000) {
                              // A-Dコンバータが使用可能に
                              // なるまで待つ.
        i++;
    }
    REFCON = 0x03;    // 外部リファレンス使用，内部リファレ
                      // ンスをVrefピンに接続
    ADCCON = 0x17A4; // fADC/32, 16 clocks、シングル・
                      // エンド・モード，連続のソフトウェア
                      // 変換
}
```

ります．

表10はタッチ・パネルとADuC7026の結線図および*XY*座標を読み込むための入出力設定です．これはマイコンよりポートを制御することで実現できます．

ADuC7026はポート0の入出力方向や出力データの設定をGP0DATというレジスタにより行います．**図12**に示すように，GP0DATレジスタの

- ビット31～24　　入出力の設定
- ビット23～16　　出力データの書き込み

に設定することにより行います．

11．A-Dコンバータの制御方法

A-Dコンバータから分圧比を読み込むための処理について説明します．今回は簡単にするため割り込みを利用して

表12　マイクロコントローラADuC7026のADCCONレジスタの概要と設定値

ビット	説明	設定値
15～13	予約	000
12～10	ADCクロック速度 000：fADC/1 001：fADC/2 010：fADC/4 011：fADC/8 100：fADC/16 101：fADC/32	101(確実のため，一番遅く設定)
9～8	ADCアクイジション時間 00：2クロック 01：4クロック 10：8クロック 11：16クロック	11(確実のため，一番遅く設定)
7	変換開始フラグ 0：無効 1：有効	1
6	ADC_{BUSY}有効フラグ 0：無効 1：有効	0(ADCBUSY端子はLEFT端子と共用のため，有効にしてはならない)
5	ADCパワー制御 0：パワーダウン・モード 1：ノーマル・モード	1
4～3	変換モード 00：シングル・エンド・モード 01：差動モード 10：擬似差動モード	00
2～0	変換タイプ 000：CONVSTARTピンを変換入力とする 001：Timer1を変換入力とする 010：Timer2を変換入力とする 011：単一のソフトウェア変換 100：連続のソフトウェア変換 101：PLA変換	100

いませんが，CPUの利用効率を向上させるには割り込みを利用することをお勧めします．

まずADuC7026のA-Dコンバータ制御に使用するレジスタの概要を**表11**に示します．今回使用するのは，ADCCON，ADCCP，ADCSTA，ADCDAT，REFCONです．

● 各種設定処理

A-Dコンバータの各種設定として，ADCCONレジスタの概要と，今回のサンプル・プログラムでの設定値を**表12**に示します．また，設定処理のソース・コードを**リスト1**に示します．

今回，簡単にするためA-D変換モードをシングル・エンド・モード，変換タイプを連続のソフトウェア変換に設定しました．

シングル・エンド・モードの場合，正チャネルとグラウンドとの電位差を取得するので，負チャネルの選択を行う必要がありません．また，連続のソフトウェア変換を行っているので，変換の開始・停止制御を行う必要がありません．

この他に，REFCONレジスタにバンドキャップ・リファレンスの設定をする必要があります．今回2.5Vの内部リファレンスではなく，3.3VのV_{REF}を使用するように設定しました．

● ADCの選択処理

画像ベースボードでは，TOP端子はADC10に，LEFT端子はADC11に繋がっています．そのため，横軸の値を読むにはADC10，縦軸の値を読むにはADC11を選択する必要があります．

これはADCCPレジスタに設定することで選択することができます．ADCCPレジスタの仕様を**表13**に示します．

● データの取得処理

A-Dコンバータからデータを取得する処理はとても簡単です．データ取得処理のソース・コードを**リスト2**に示します．

ADCSTAレジスタのADCReadyフラグ(0ビット目)がONしたら変換完了なので，ADCDATレジスタの値を取得します．ADCDATレジスタから取得するデータの形式を**図13**に示します．

GPIOの設定変更もしくはA-Dコンバータの設定変更直後の取得データは不安定になることがあります．**リスト2**のサンプル・プログラムでは抵抗膜駆動電圧切り替え後の安定のため，4回読み捨てを行い，5回目の取得値を採用しています．

表13　マイクロコントローラADuC7026のADCCPレジスタの仕様

ビット	説　明
7-5	予約
4-0	正チャネル選択ビット
	00000：ADC0
	00001：ADC1
	00010：ADC2
	00011：ADC3
	00100：ADC4
	00101：ADC5
	00110：ADC6
	00111：ADC7
	01000：ADC8
	01001：ADC9
	01010：ADC10
	01011：ADC11
	01100：DAC0/ADC12
	01101：DAC1/ADC13
	01110：DAC2/ADC14
	01111：DAC3/ADC15
	10000：温度センサ
	10001：AGND(自己診断機能)
	10010：内部リファレンス(自己診断機能)
	10011：$AV_{DD}/2$

リスト2　データの取得処理

```
short AD_get_data(int num)
{
    short data, tmp;
    int i;

    ADCCP = num;    // 使用するA-Dコンバータの選択

    for (i=0; i<5; i++) { //  4回分読み捨て，最後のデータ
                          // のみ採用
        while (!ADCSTA);    // A-D変換完了まで待つ

        tmp = ADCDAT >> 16; // データ読み込み
    }

    data = tmp & 0x0FFF;    // ADC 12ビット・リザルト
    if (tmp & 0xF000) {     // 符号ビット
        data *= (-1);
    }
    return data;
}
```

31　28	27　16	15　0
符号ビット	12ビットのADCリザルト	未使用

図13　ADCDATレジスタのデータ形式

リスト3　タッチ・パネルの位置座標算出プログラム(tpn1_common.c)

```
#define WIDTH   (640)
#define HEIGHT  (480)

#define POS_X1  (0)         // (X1,Y1)-----(X2,Y1)
#define POS_Y1  (0)         // |                 |
#define POS_X2  (WIDTH-1)   // |                 |
#define POS_Y2  (HEIGHT-1)  // (X1,Y2)-----(X2,Y2)

#if 1 /* 4点算出版 */

#define AD_X1   (450)      // AD1-------AD2
#define AD_Y1   (590)      // |           |
#define AD_X2   (3600)     // |           |
#define AD_Y2   (680)      // AD3-------AD4
#define AD_X3   (450)
#define AD_Y3   (3400)
#define AD_X4   (3600)
#define AD_Y4   (3400)

void TPNL_get_pos(short ad_x, short ad_y, short *pos_x,
short *pos_y)
{
    long pos_xa, pos_xb, tmp_pos_x;
    long pos_ya, pos_yb, tmp_pos_y;
    // (X1,Y1)-----(XA,Y1)-----(X2,Y1)
    // |             |               |
    // (X1,YA)----- (X,Y) -----(X2,YB)
    // |             |               |
    // (X1,Y2)-----(XB,Y2)-----(X2,Y2)

    pos_xa = ((POS_X2 - POS_X1) * ad_x + POS_X1 * AD_X2
                                       - POS_X2 * AD_X1)
                / (AD_X2 - AD_X1);
    pos_xb = ((POS_X2 - POS_X1) * ad_x + POS_X1 * AD_X4
                                       - POS_X2 * AD_X3)
                / (AD_X4 - AD_X3);

    pos_ya = ((POS_Y2 - POS_Y1) * ad_y + POS_Y1 * AD_Y3
                                       - POS_Y2 * AD_Y1)
                / (AD_Y3 - AD_Y1);
    pos_yb = ((POS_Y2 - POS_Y1) * ad_y + POS_Y1 * AD_Y4
                                       - POS_Y2 * AD_Y2)
                / (AD_Y4 - AD_Y2);

    if (pos_xa == pos_xb) {
        tmp_pos_x = pos_xa;
    }
    else {
        tmp_pos_x = ((pos_ya - pos_yb) * POS_X1 *
                                       (pos_xb - pos_xa)
                        - POS_Y2 * pos_xa)
                    * (POS_X1 - POS_X2)
                    / ( (POS_X2 - POS_X1) * (POS_Y2
                                               - POS_Y1)
                        - (pos_xb - pos_xa) * (pos_yb -
                                               pos_ya) );
    }
        tmp_pos_y = (pos_yb - pos_ya) * tmp_pos_x / (POS_X2
                                          - POS_X1) + pos_ya;

        if (tmp_pos_x < POS_X1) {
            tmp_pos_x = POS_X1;
        }
        else if (tmp_pos_x > POS_X2) {
            tmp_pos_x = POS_X2;
        }
        if (tmp_pos_y < POS_Y1) {
            tmp_pos_y = POS_Y1;
        }
        else if (tmp_pos_y > POS_Y2) {
            tmp_pos_y = POS_Y2;
        }

        *pos_x = (short)tmp_pos_x;
        *pos_y = (short)tmp_pos_y;
}

#else /* 簡易算出版 */

#define AD_X1   (450)       // (X1,Y1)-----(X2,Y1)
#define AD_Y1   (600)       // |                 |
#define AD_X2   (3600)      // |                 |
#define AD_Y2   (3400)      // (X1,Y2)-----(X2,Y2)

void TPNL_get_pos(short ad_x, short ad_y, short *pos_x,
                                             short *pos_y)
{
    long tmp_pos_x, tmp_pos_y;

    tmp_pos_x = ((POS_X2 - POS_X1) * ad_x + POS_X1 *
                                 AD_X2 - POS_X2 * AD_X1)
                / (AD_X2 - AD_X1);
    if (tmp_pos_x < POS_X1) {
        tmp_pos_x = POS_X1;
    }
    else if (tmp_pos_x > POS_X2) {
        tmp_pos_x = POS_X2;
    }

    tmp_pos_y = ((POS_Y2 - POS_Y1) * ad_y + POS_Y1 *
                                 AD_Y2 - POS_Y2 * AD_Y1)
                / (AD_Y2 - AD_Y1);
    if (tmp_pos_y < POS_Y1) {
        tmp_pos_y = POS_Y1;
    }
    else if (tmp_pos_y > POS_Y2) {
        tmp_pos_y = POS_Y2;
    }

    *pos_x = (short)tmp_pos_x;
    *pos_y = (short)tmp_pos_y;
}

#endif
```

12．タッチ位置座標の取得

A-Dコンバータより取得できた値をもとに，座標データの算出を行います．タッチ・パネル自体の個体差もあるため，A-Dコンバータより取得できる値はタッチパネルごとに異なります．そのため**図14**のように，初期設定の際に4点のタッチ情報を入力するのが一般的です．

今回のサンプル・プログラムでは，あらかじめ4隅の値を計測し，プログラム中では固定で定義しているので，必要に応じて変更します(**リスト3**．lib/tpnl_common.c)．

タッチ位置座標(POS_X, POS_Y)の算出には以下の情報が必要になります．

- 初期設定4点の座標情報(POS_X1, POS_X2, POS_Y1, POS_Y2)
- 初期設定4点のA-Dコンバータ情報(AD_X1～AD_X4, AD_Y1～AD_Y4)
- タッチ位置が同じX座標でも，Y座標が違えば，A-Dコンバータ情報は異なるので，注意が必要．
- タッチ位置のA-Dコンバータ情報(AD_X, AD_Y)

図14 初期設定の際にタッチ・パネルの4点の情報を入力する
初期設定の際に4点のタッチ情報を入力する．

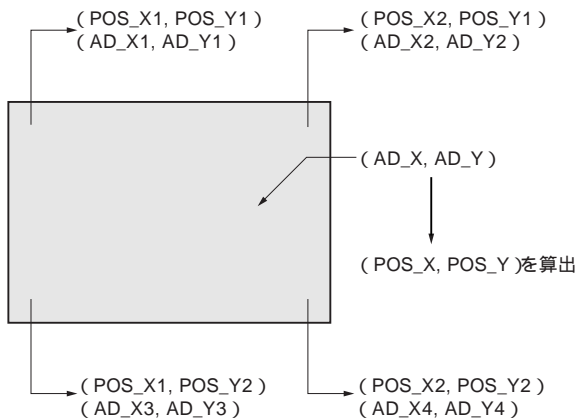



図15 タッチ情報の関係
タッチパネルの押された*XY*座標値を，初期設定で確認した4隅の基準点より算出．

●タッチ・パネルの解像度(WIDTH，HEIGHT)

これらの情報の関係を**図15**に示します．

タッチ位置座標は，これらの情報を使用して連立方程式を解くことで求めることができます．算出式は非常に複雑なため，誌面の都合上割愛します．

興味のある方は**リスト3**(tpnl_common.c)のソース・コードを参照ください．

サンプル・プログラムでは，算出した座標情報をシリアル出力しているので，パソコンのシリアル端末などで表示することができます．出力フォーマットを**図16**に示します．

● おわりに

タッチ・パネル液晶の座標の取得方法について紹介しました．またこのプログラムを使って**図9**に示したシステムを実現するプログラムmain.cを**リスト4**に示します．

マイコン，FPGA，タッチ・パネル液晶，カメラなどを連携することで，より高機能なシステムを開発できると思います．その機能の一部を実現するための参考になれば幸いです．

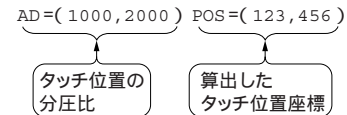



図16 座標情報の出力フォーマット

リスト4 図9のシステムを実現するプログラム(main.c)

```
#include "lib/libuart.h"
#include "lib/libad.h"
#include "lib/libtpnl.h"

int main (void)
{
    short ad_x, ad_y;
    short pos_x, pos_y;

    AD_init();
    UART_init();
    TPNL_init();

    while(1) {
        ad_x = TPNL_get_ad_x();
        ad_y = TPNL_get_ad_y();
        TPNL_get_pos(ad_x, ad_y, &pos_x, &pos_y);

        UART_put_str((unsigned char *)"AD=(");
        UART_put_num(ad_x);
        UART_put_str((unsigned char *)",");
        UART_put_num(ad_y);
        UART_put_str((unsigned char *)")");

        UART_put_str((unsigned char *)" POS=(");
        UART_put_num(pos_x);
        UART_put_str((unsigned char *)",");
        UART_put_num(pos_y);
        UART_put_str((unsigned char *)")\n");
    }

    return 0;
}
```

参考・引用*文献

(1)* 京セラ；TCG057VGLAC-G00仕様書，2006年6月．
(2)* National Semiconductor；LM2733データシート，2003年2月．

えさき・まさやす
㈱イーエスピー企画
もりかわ・あきひさ
㈱ヴィッツ

<筆者プロフィール>
森川聡久．組み込みソフトウェア・エンジニア．主に車載プラットフォームや情報家電の開発を行っている．しかし，最近は管理業務や社員教育の割合が増えてきているので，週末にイーエスピー企画 土日システム開発部に所属し，開発技術を磨いている．SWEST実行委員も務める．